第9章　単体法

線形計画法は古くから研究されており、理論的にも実用的にも大きな成果をあげている。その成功の理由のひとつは、効率的な解法である単体法の開発による。単体法は、凸多面体集合で表された集合の端点を効率良く探索し、線形計画問題の解を見つける手法である。

9.1　単体法の概略

この節では、単体法の概略を説明する．

単体法は，線形計画問題の線形制約が多面体になっており、その解が多面体の端点となるという事実を巧みに使って、解を求める手法である (図 9.1)。

そこで，この節では，以下のように議論を進めていくことにする．

1. 標準形で表さされた線形制約 $Ax = b, x \geq 0$ の端点とは，どんな点であるか？
   線形制約で表される凸多面体の端点は、行列 A を用いてどのように表すことができるかを見る．
2. 端点をどのように調べるか？
   一般に凸多面体の端点は次元 n の指数オーダー存在する。そのため、端点をすべて調べることは事実上不可能である。そこで端点の効率がよい探索の仕方である単体法を紹介する．
3. 単体法をどのように実装するか？
   単体法を実際に行なうときに効率よい計算手法であるシンプレックスタブローを使った方法を紹介する．

9.1.1　標準形で表さされた線形制約の端点は，どのような点か？

線形計画問題を章でも見たように、線形計画問題の解は線形制約の端点で達成される。そこで、端点をしらみつぶしに調べれば，線形計画問題の解を得ることができる．そこで，端点はどのように計算できるか見てみよう．

線形計画問題を章では，線形制約集合が次のような線形不等式だけで表されている場合を考えていた。

$$p_i^T x \leq q_i \;\; i = 1, \ldots, r$$

このとき、この制約で表される多面体の端点は、$p_i, i \in B$ が線形独立となる n 個の方程式

$$p_i^T x = q_i, i \in B$$

の解であると定義していた。ここで B は n 個の要素からなる添え字集合である。

それでは、線形計画問題の制約集合が標準形

$$a_i^T x = b_i, \; i = 1, \ldots, m, \; x \geq 0$$

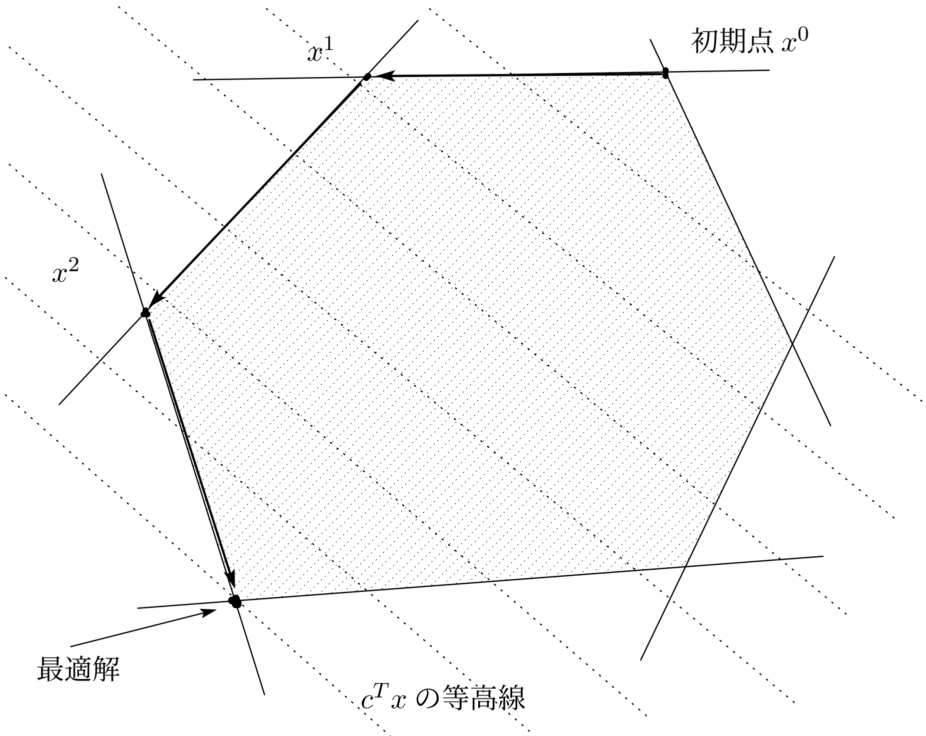

図 9.1: シンプレックス法のイメージ

で表されている場合の端点はどのように与えられるだろうか？この場合も、この条件を不等式制約に変換してやることによって、特徴付けることができる。ここで、$\alpha = 0$ は、$\alpha \geq 0$ かつ $\alpha \leq 0$ であることと同じであることに注意しよう。この性質を用いれば、標準形は、

$$a_i^T x \geq b_i,\ a_i^T x \leq b_i,\ i = 1, \ldots, m,\ x \geq 0$$

というすべて不等式の制約で表すことができる。このことより、標準形の端点は、a_i, e_i から n 個の 1 次独立のベクトルをとってきて、その方程式の解と定義することができる。(ここで e_i は制約 $x_i \geq 0$ が $(e_i)^T x \geq 0$ となることによる。)

$$a_i^T x = b_i \;\; i \in \bar{B}, x_j = 0 \;\; j \in \bar{N}$$

ここで、$\bar{B}$, $\bar{N}$ はその要素の和が n である添え字集合である。このように定義される端点の中から、等式条件を満たす端点を考えよう。この端点、$Ax = b$ であるため、端点を形成する 1 次独立なベクトルには、m 個の $\{a_i\}$ が選ばれ、残りの $n - m$ 個のベクトルは $\{e_j\}$ から選ばれる。このとき、e_j が選ばれた j に対しては $x_j = 0$ となる。このような端点を線形計画問題の基底解と呼ぶ。

基底解 x では，x の成分のうち $n - m$ 個が 0 となり，他の成分が非負となるもので表されているということが言える．つまり，基底解は，$n - m$ 個のゼロとなる要素の集合 N と N の補集合 B を用いて，

$$A_B x_B = b, x_B \geq 0, x_N = 0 \tag{9.1}$$

と表すことができる．ここで，A_B は，添字 $i \in B$ に対応した A の列を抽出した行列であり，x_B, x_N はそれぞれ，B, N に対応した成分を抜き出した m 次元ベクトルと $n - m$ 次元ベクトルである．このとき，x_B を基底変数，x_N を非基底変数と呼ぶ．ここで、基底解は、$x_B \leq 0$ という保証がないため、かならずしも実行可能解ではないことに注意しよう。そこで、基底解の中でも実行可能なものを、特に実行可能基底解と呼ぶことにする。

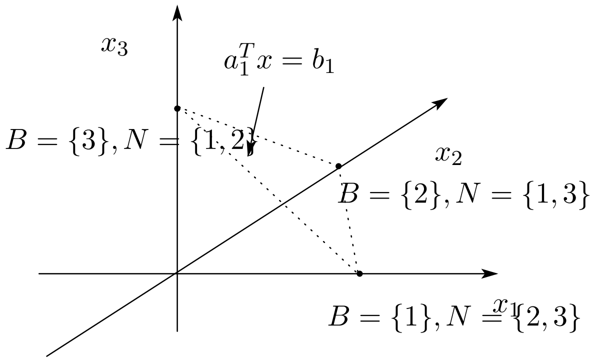

図 9.2: 隣あう端点の添え字集合 B, N の関係

線形計画問題の実行可能集合である多面体の端点は (9.1) として表されていることになるので，B と N の組み合せの数 C_n^m だけある端点を調べれば、最適解を見つけることができる。しかし，n が大きくなればなるほど，この数は一般に n の指数オーダーで大きくなる．そこで，効率よく端点を調べる方法を考える必要がある．

9.1.2 端点をどのように調べるのか？

効率よく端点を調べたらよいのだろうか？多面体の図を見ると明らかなように，端点とは互いに繋がっている．そこで，繋がっている端点の中から目的関数の値 $c^T x$ を小さくしそうな端点を検索すればよいように思える．

それでは，隣り合っている端点とは何か？これは，実は，B, N が与えられているとき，その中から 1 つずつ取り出して，入れ替えたものに対応している (図 9.2).

そこでどの添字を入れ替えたらよいのか考えよう．現在の目的関数値は，

$$c^T x = c_B^T x_B + c_N^T x_N$$

となる。今、x が実行可能基底解であるとすると、

$$A_B x_B + A_N x_N = b$$

であるので、x_B は、

$$x_B = A_B^{-1} b - A_B^{-1} A_N x_N$$

となる。(実際は $x_N = 0$ であるので、$x_B = A_B^{-1} b$ であることに注意) この式を代入すると、

$$c^T x = c_B^T A_B^{-1} b + (c_N^T - c_B^T A_B^{-1} A_N) x_N$$

と表される。今、$x_N = 0$ であるが、隣の実行可能基底解に移動するためには、集合 N の中から添え字 j 選び、$x_j > 0$ となるようにする必要がある。(j 以外の x_N は 0 のままである。) しかしながら、

$$c_N^T - c_B^T A_B^{-1} A_N \geq 0$$

であれば、そのような変化に対して目的関数値が増加してしまうことを意味している。このことは、現在の基底解 x が、この線形計画問題の局所的最適解であることを意味している。線形計画問題は凸計画問題であったので、局所最適解は大域的最適解になる。つまり、実行可能基底解 x

は $c_N^T - c_B^T A_B^{-1} A_N \geq 0$ であるとき、最適解となる。この $c_N^T - c_B^T A_B^{-1} A_N$ を基底解の相対コスト係数と呼ぶ。ここで、相対コスト係数を $\lambda = c_N^T - c_B^T A_B^{-1} A_N$ とする。

ここで行ベクトルであらわされている相対コスト係数 λ のなかで、$\lambda_i < 0$ になる添え字 $i \in N$ があれば、その添え字に対応する非基底変数 x_i を基底変数にし、非基底変数の中からひとつ非基底変数になるよう、新たに実行可能な基底解を求めてやれば、次の端点が求まる。ここで探索方向 d は、i 番目の要素が 1 でその他の要素が 0 となる $n-m$ 次元ベクトルを e_i とすると、

$$d = \begin{pmatrix} -A_B^{-1} A_N e_i \\ e_i \end{pmatrix}$$

であり、この方向は、

$$\nabla f(x)^T d = c_N^T(-A_B^{-1} A_N e_i) + c_B^T e_i = \lambda_i < 0$$

となり降下方向となっていることがわかる。この観点から、$\lambda_i < 0$ となる i の中でも、一番小さいものを選ぶと、目的関数の値が大きく減らすことができると期待できる。目的関数は線形であるので、正であるどんなステップサイズ t でも目的関数は下がる。

$$c^T x > c^T (x + td)$$

そこで、次にステップサイズ t を決めるわけだが、t は以下を満たさなければならない。

- $x + td$ が制約を満たすこと、つまり $x + td \geq 0, A(x + td) = 0$ となる。
- $x + td$ が端点になる、つまり、$(x + td)_B$ のうちひとつは 0 になる。

まず、一つ目だが、a_i を行列 A の i 番目の列を表すとすると、

$$A(x + td) = Ax + tA \begin{pmatrix} -A_B^{-1} A_N e_i \\ e_i \end{pmatrix} = b + t(-a_i + a_i) = b$$

となる。ここで、$A_N e_i = a_i$ であることを用いている。この式より t に関係なく等式制約 $Ax = b$ をみたすことがわかる。次に、不等式制約を満たす条件を考える。N に対応した要素では、$x_N + te_i$ であるため、$t \geq 0$ であるので明らかに満たす。そこで、B に対応する要素を考える.

$$x_B + t(-A_B^{-1} A_N e_i) = x_B - tA_B^{-1} a_i$$

となる。$x_B \geq 0$ であることに注意すると、もし $A_B^{-1} a_i$ のすべての要素が 0 以下であれば、任意の $t \geq 0$ に対して、$x_B - tA_B^{-1} a_i \geq 0$ となる。つまり t に関係なく制約条件が満たされることになる。そのため、t を大きくすれば、目的関数はいくらで小さくすることができる。つまりこの問題は有界でないということになる。

もし、$A_B^{-1} a_i$ の中で正となる要素があれば、t を大きくしていくと、いつか、

$$(x_B - tA_B^{-1} a_i)_j = 0$$

となる要素 j が出てくる。この $j \in B$ に対応する x_j が次の端点における非基底変数となる。このとき、ステップサイズ t は、

$$t = \min_{j \in B} \left\{ \frac{(x_B)_j}{(B^{-1} a_i)_j} \mid (B^{-1} a_i)_j \geq 0 \right\}$$

となる。

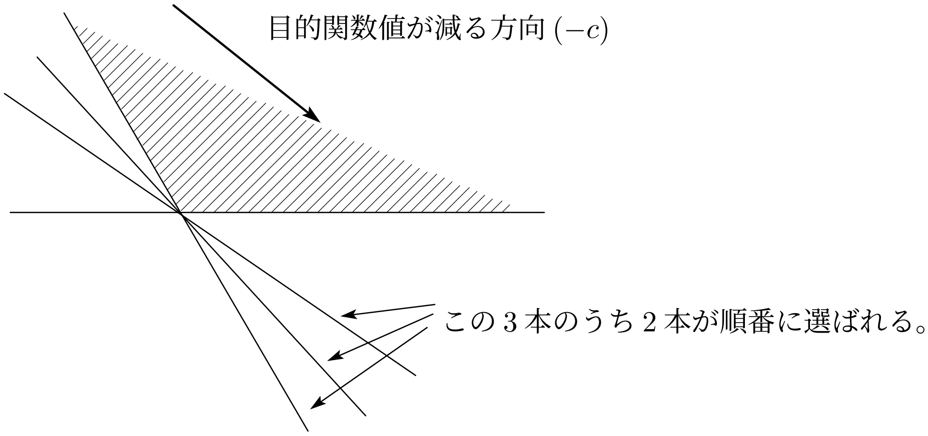

図 9.3: ステップサイズが 0 になるとき

9.2　単体法

ここでは、前節で説明した単体法の仕組みをまとめて、きちんとした記述を与え、その性質を紹介する。

前節のことをまとめるとシンプレックス法は以下のように記述できる。

Algorithm 9.2.1 ステップ 0 初期実行可能解 $(x_B, x_N) = (A_B^{-1}b, 0)$ を選ぶ。

ステップ 1　相対コスト係数 $\lambda = c_N^T - c_B^T A_B^{-1} A_N$ を計算する。

ステップ 2　$\lambda \geq 0$ であれば今の解が最適解となり終了する。そうでなければ、λ_i が最小となる i を選ぶ。

ステップ 3 $A_B^{-1}a_i$ に正の要素がなければ、問題は有界でないので終了。そうでなければ、ステップサイズを以下のように計算する。

$$t = \min_{j \in B} \left\{ \frac{(x_B)_j}{(B^{-1}a_i)_j} \mid (B^{-1}a_i)_j > 0 \right\}$$

ここで上式の最小の値を与える添え字 $j \in B$ をひとつ選び、$\bar{j}$ とする。

ステップ 4 新しい端点を、

$$x = x + t \begin{pmatrix} -B^{-1}Ne_i \\ e_i \end{pmatrix}$$

とする。B から $\bar{j}$ を抜き、i を加えたものを新しい B とする。N から i を抜き、$\bar{j}$ を加えたものを新しい N とする。ステップ *1* へ。

実行可能な端点の数は有限個であるので、単体法においてステップサイズが 0 とならなければ、有限解で解を得ることができる。ほとんど稀なケースではあるが、ステップサイズが最適解ではないにもかかわらず 0 になることがる。これは、端点において $n+1$ 個以上の $a_i^T x = b_i$ が成り立つときに起こる (図 9.3).

そのようなときには、何回も繰り返すと B, N が同じものになることがある。そこで、そのような B, N が同じにならないようにステップ 2 において添え字を選ぶ手法として、辞書式順序法が提案されている。

基本的な単体法は以上のものであったが、これを計算するときに便利な手法を次の節で与える。

9.3 シンプレックスタブロー

この説では、単体法を実装する上で便利な、シンプレックスタブローを紹介し、具体的にどのようにシンプレックスタブローを用いるか説明する。単体法を計算する上で、シンプレックスタブローと呼ばれる表を用いると便利である。

単体法のアルゴリズムで見たように、シンプレックス法で重要な役割をはたす式は、基底変数の添え字集合 B, 相対コスト係数 λ, $A_B^{-1}a_j, j = 1,\ldots,n$, $x_B = B^{-1}b$ であることがわかる。そこで、これらを表にしてみることにする。

- 一番左の列に基底変数
- 一番上の行には、B に対応するところには 0, N に対応するところには相対コスト係数。
- 一番上の右端に、目的関数の値に -1 を掛けたもの。
- 相対コスト係数の下に、$B^{-1}a_j$
- 一番右の列に $B^{-1}b$

$$\Downarrow$$

	$0 \ \lambda$				$-c^Tx$
x_B	$B^{-1}a_1$	$B^{-1}a_2$	$\cdots$	$B^{-1}a_n$	$B^{-1}b$

この表の一番上の行を見ることによって、現在の状況がわかる。もし相対コスト係数がすべて 0 以上なら、その点が最適解であるので終了する (ステップ 2)。そうでなければ、どれかひとつは負になっているはずであるので、そのなかからひとつの添え字 i を選ぶ。次に、右の列 $A_B^{-1}b$ をチェックする。すべて 0 以下なら、この問題は有界でないので終了 (ステップ 3)。そうでなければ $A_B^{-1}b > 0$ の j 行と i 列に着目して、$\frac{(x_B)_j}{(A_B^{-1}a_i)_j}$ をすべて計算する。その中で、一番大きい j を $\bar{j}$ とする (ステップ 3)。このとき、非基底変数 i と基底変数 $\bar{j}$ を入れ替えることになる (ステップ 4)。この入れ替えをピボットと呼ぶ。これだけでは、単にアルゴリズムを説明しただけだが、この表を用いると実は簡単に次の端点に対するシンプレックスタブローが作ることができるのである。ここで、シンプレックスタブローにおける $A_B^{-1}a_1 \quad A_B^{-1}a_2 \cdots A_B^{-1}a_n$ の中で、$\bar{j}$ 行 i 列に対応する値をピボット要素と呼ぶ。まず、シンプレックスタブローのピボット要素があるところの行をすべて、ピボット要素で割る（この計算量は $O(m)$)。できあがった行ベクトルを $\bar{a}$ とする。なお、このとき、ピボット要素のあるところの値が 1 になる。次に、さらにピボット要素があった行以外の行は、ピボット要素があったところの列の値が 0 になるように、$\bar{a}$ を適当に何倍かしたものを足す (引く)。(この操作は線形方程式のガウスの消去法に似ている．実際、新しい基底変数に対応するところが単位行列になるような操作である。) この操作は、一番上の相対コスト係数、目的関数値に -1 を掛けたものにも施してやる (この計算量は $O(mn)$)。すると、この操作は、新しい基底変数に対するシンプレックスタブローになっている。(このシンプレックスタブローは $\bar{j}$ のあったところが、i に置き換わった表になっていることに注意)

前節のアルゴリズムを単純に実行すると、B^{-1} の計算をする必要があり、その計算量は $O(n^3)$ である。シンプレックスタブローを使えば、その計算量は $O(mn)$ であるので、シンプレックスタブローを使う方が速く実行できることがわかる。

それでは、実際にシンプレックスタブローを使って、次の問題を解いてみよう。

$$\begin{array}{ll} \min & -x_1 - x_2 \\ \text{subjectto} & 3x_1 + 2x_2 + x_3 = 12 \\ & x_1 + 2x_2 + x_4 = 8 \\ & x \geq 0 \end{array}$$

この問題の実行可能基底解のひとつは $(0, 0, 12, 8)^T$ であり、$B = \{3, 4\}$ である。これを初期点としたシンプレックスタブローは、

	-1	-1	0	0	0
x_3	3	2	1	0	12
x_4	1	2	0	1	8

となる。相対コスト係数は $i = 1, 2$ のところが負になっている。ここでは、$i = 1$ とする。

次に右端の列を着目するとすべて正であるので、$\{\frac{(x_B)_j}{(A_B^{-1} a_i)_j}$ を計算する。この表では、$j = 1$ のところで $\frac{1}{4}$, $j = 2$ のところで $\frac{1}{8}$ となる。$j = 1$ の方が大きいので、$\bar{j} = 1$ となる。つまりピボット要素は 3 となる。そこで、x_3 の行をすべて 3 で割る。この計算された行は $(1, 2/3, 1/3, 0, 4)$ となる。これを、1 倍したものを 1 番上の行に足し、1 倍したものを一番下の行から引くと以下のシンプレックスタブローを得る。

	0	-1/3	1/3	0	4
x_1	1	2/3	1/3	0	4
x_4	0	2	1/3	1	4

となる。このとき、相対コスト係数が、$i = 2$ のところが負であるので、ピボットを行う。すると次の表を得る。

	0	0	1/4	1/4	5
x_1	1	0	1/2	-1/2	2
x_2	0	1	-1/4	1/4	3

この表では、相対コスト係数がすべて非負であるので、最適解が得られていることを示している。つまり、基底変数 x_1, x_2 は一番右端の値になっており、非基底変数では $x_3 = x_4 = 0$ である。さらにこのときの目的関数値は、一番右上の値に -1 を掛けたもの、つまり -5 となる。

今のシンプレックスタブローの計算では、あらかじめ初期実行可能基底解が求まっていたことになっていたが、この初期実行可能基底解を求めることはそれほど簡単な問題ではない。そこで次の節では、その初期実行可能基底解を求める方法を説明する。

9.4 2 段階法

単体法では初期点として、初期実行可能基底解を求める必要があった。ここで、初期実行可能基底解を求める手法を説明する

線形計画問題:

$$\begin{array}{ll} \min & c^T x \\ \text{subjectto} & Ax = b \\ & x \geq 0 \end{array}$$

の初期実行可能基底解を求めるために、次の性質を持つ人為的な線形計画問題を作る。

- 初期実行可能基底解が簡単に求まる。
- 最適解が解きたい問題の初期実行可能基底解となる。

そのような問題として、

$$\begin{array}{ll} \min & \sum_{j=1}^{m} y_j \\ \text{subjectto} & (Ax)_j + (\text{sign}(b_j))y_j = b_j \quad j = 1, \ldots, m \\ & x \geq 0, y \geq 0 \end{array}$$

が考えられる。ここで、sign(α) は、$\alpha \geq 0$ のとき 1, $\alpha < 0$ のとき -1 を返す関数である。元の問題が実行可能であれば、この問題の最適解は $y = 0$ となる。さらに、$x = 0, y = b$ とすると、これは実行可能基底解となる。そして、これを初期実行可能基底解にしてシンプレックス法を実行して得られた解が、元の問題の実行可能基底解となる。この操作を第 1 段階といい、第 1 段階で得られた実行可能基底解を用いて元の問題を解く操作を第 2 段階と呼ぶ。そのため、このような手法を 2 段階法と呼ぶ。